WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Thursday, June 09, 2005

Hide?	Set Name	Query	Hit Count
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
☐	L11	L7 and L1	2
	DB=EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
☐	L10	L4	5
☐	L9	L3	314
☐	L8	L7	0
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
☐	L7	L6 and programmable	505
☐	L6	L5 and assembly	671
☐	L5	L4 and counter	1580
☐	L4	L3 and (analog-to-digital or ADC)	2610
☐	L3	L2 and timer	11739
☐	L2	micro?controller or microcontroller	59076
☐	L1	717/106-113.ccls.	1270

END OF SEARCH HISTORY

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 2 of 2 returned.

☐ 1. Document ID: US 20040005859 A1

L11: Entry 1 of 2 File: PGPB Jan 8, 2004

PGPUB-DOCUMENT-NUMBER: 20040005859
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20040005859 A1

TITLE: Wireless deployment / distributed execution of graphical programs to smart sensors

PUBLICATION-DATE: January 8, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Ghercioiu, Marius	Austin	TX	US	
Ceteras, Ciprian	Baia Mare	TX	RO	
Monoses, Ioan	Gherla jud Cluj		RO	
Crisan, Gratian I.	jud Cluj		RO	
Kodosky, Jeffrey L.	Austin		US	

US-CL-CURRENT: 455/3.01; 717/109

ABSTRACT:

System and method for deploying or executing a graphical program to a device in a wireless manner. A graphical program (GP) is created that implements a measurement function. Some or all of the GP is transmitted to a hub over a network. The hub executes the transmitted GP and sends corresponding commands to a measurement device via wireless means in accordance with a wireless communication protocol. The measurement device executes the commands to perform the measurement function, thereby generating resultant data, which is received from the measurement device via wireless means. The GP may include a block diagram that executes on the measurement device, and a user interface portion that is displayed by a first computer system. Transmitting the GP to the hub may include generating a machine-executable program based on the GP and transmitting the machine-executable program to the hub for execution.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

☐ 2. Document ID: US 5852733 A

L11: Entry 2 of 2 File: USPT Dec 22, 1998

US-PAT-NO: 5852733
DOCUMENT-IDENTIFIER: US 5852733 A

TITLE: Microcontroller development tool using software programs

DATE-ISSUED: December 22, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Chien; Yung-Ping S.	Indianapolis	IN	46202	
Connor; William D. S.	Pendleton	IN	46064-9541	
Jeffares; Christopher D.	Noblesville	IN	46061	

US-CL-CURRENT: 717/113; 712/201, 717/114

ABSTRACT:

A software development tool is provided for Texas Instruments microcontrollers which provides register initialization, register information, and register editing in conjunction with Windows.RTM.-based forms generated by programs written according to well-known software programming languages. The editor function of the software development tool functions without the use of an emulator.

20 Claims, 9 Drawing figures
Exemplary Claim Number: 5
Number of Drawing Sheets: 9

1 ITiCSE 2000 working group reports: Distributed expertise for teaching computer organization & architecture

Lillian (Boots) Cassel, Mark Holliday, Deepak Kumar, John Impagliazzo, Kevin Bolding, Murray Pearson, Jim Davies, Gregory S. Wolffe, William Yurcik

June 2001 **Working group reports from ITiCSE on Innovation and technology in computer science education**

Full text available: pdf(1.89 MB) Additional Information: full citation, abstract, references, citings, index terms

This report presents preliminary results from our project on creating distributed expertise for teaching computer organization & architecture course(s) in the undergraduate computer science curriculum. We present the details of an online survey designed to gather information from faculty on the current state of teaching this course. The survey also tries to identify specific areas of need for creating distributed expertise as reported by various faculty. We also present several resources that ha ...

2 ITiCSE 2000 working group reports: Distributed expertise for teaching computer organization & architecture

Lillian (Boots) Cassel, Mark Holliday, Deepak Kumar, John Impagliazzo, Kevin Bolding, Murray Pearson, Jim Davies, Gregory S. Wolffe, William Yurcik

June 2001 **ACM SIGCSE Bulletin**, Volume 33 Issue 2

Full text available: pdf(1.89 MB) Additional Information: full citation, abstract, references, citings

This report presents preliminary results from our project on creating distributed expertise for teaching computer organization & architecture course(s) in the undergraduate computer science curriculum. We present the details of an online survey designed to gather information from faculty on the current state of teaching this course. The survey also tries to identify specific areas of need for creating distributed expertise as reported by various faculty. We also present several resources that ha ...

3 Teaching microcontrollers with hands-on hardware experiments

Rafic Bachnak

April 2005 **Journal of Computing Sciences in Colleges**, Volume 20 Issue 4

Full text available: pdf(444.48 KB) Additional Information: full citation, abstract, references

The Department of Computing and Math Sciences at Texas A&M University-Corpus Christi has developed a Digital Systems Laboratory that affords Computer Science and Engineering Technology students state-of-the-art training tools. This paper presents the major components of the laboratory and describes one of the hands-on software and hardware experiments developed for teaching a senior-level course in microprocessors and microcontrollers.

4 Compiling with code-size constraints

Mayur Naik, Jens Palsberg

February 2004 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 3 Issue 1

Full text available: pdf(178.97 KB) Additional Information: full citation, abstract, references, index terms, review

Most compilers ignore the problems of limited code space in embedded systems. Designers of embedded software often have no better alternative than to manually reduce the size of the source code or even the compiled code. Besides being tedious and error prone, such optimization results in obfuscated code that is difficult to maintain and reuse. In this paper, we present a step towards code-size-aware compilation. We phrase register allocation and code generation as an integer linear programming p ...

Keywords: Banked architecture, integer linear programming, register allocation, space optimization

5 PLATO - PLA Translator/Optimizer - "a ROM is a PLA in no uncertain terms."

Will Sherwood

February 1977 **Proceedings of the Symposium on Design Automation and Microprocessors**

Full text available: pdf(388.86 KB) Additional Information: full citation, abstract, references, index terms

PLATO (Programmed Logic Array Translator/Optimizer) is an extremely simple system that takes mnemonic definitions and equations, assembles and optimizes them, and produces formatted output files for various uses. It was designed with human engineering goals in mind to permit painless usage and to ease the burden on the designer by relieving him of clerical tasks. PLATO was written in APL-10 because of its powerful matrix and character manipulation operators, and quick ...

Keywords: Cad techniques, Control logic, Gate arrays, Logic equations, Microcontroller, Microprocessor emulation, Optimization, PLA, ROM, Simulation, Sum of products

6 Embedded microcomputers: how to easily add it to the core computer science curriculum

Vance E. Poteat

December 2003 **Journal of Computing Sciences in Colleges**, Volume 19 Issue 2

Full text available: pdf(31.46 KB) Additional Information: full citation, abstract, references, index terms

Computer Science students today are typically well versed in software techniques and tools by the time they graduate. They are able to create solutions to most any problem as long as it can be run on standard hardware and operating systems. These types of solutions require high performance microprocessors such as the Pentium Processor and the need for such talent is evident. However, in today's world more and more products are using embedded microprocessors (aka: single chip microcomputers) and ...

7 Developing for the Atmel AVR microcontroller on Linux

Pat Deegan

February 2005 **Linux Journal**, Volume 2005 Issue 130

Full text available: html(25.18 KB) Additional Information: full citation, abstract

Make your microcontroller projects go smoothly with familiar, powerful GNU tools.

8 Polygon rendering on a stream architecture

John D. Owens, William J. Dally, Ujval J. Kapasi, Scott Rixner, Peter Mattson, Ben Mowery

August 2000 **Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on**

Graphics hardware

Full text available: pdf(161.65 KB) Additional Information: full citation, abstract, references, citings, index terms

The use of a programmable stream architecture in polygon rendering provides a powerful mechanism to address the high performance needs of today's complex scenes as well as the need for flexibility and programmability in the polygon rendering pipeline. We describe how a polygon rendering pipeline maps into data streams and kernels that operate on streams, and how this mapping is used to implement the polgyon rendering pipeline on Imagine, a programmable stream processor. We compare our resul ...

Keywords: OpenGL, SIMD, graphics hardware, kernels, media processors, polygon rendering, stream architecture, stream processing, streams

9 Exploiting dual data-memory banks in digital signal processors

Mazen A. R. Saghir, Paul Chow, Corinna G. Lee

September 1996 **Proceedings of the seventh international conference on Architectural support for programming languages and operating systems**, Volume 31 , 30 Issue 9 , 5

Full text available: pdf(1.24 MB) Additional Information: full citation, abstract, references, citings, index terms

Over the past decade, digital signal processors (DSPs) have emerged as the processors of choice for implementing embedded applications in high-volume consumer products. Through their use of specialized hardware features and small chip areas, DSPs provide the high performance necessary for embedded applications at the low costs demanded by the high-volume consumer market. One feature commonly found in DSPs is the use of dual data-memory banks to double the memory system's bandwidth. When coupled ...

10 Why SpecInt95 should not be used to benchmark embedded systems tools

Jakob Engblom

May 1999 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1999 workshop on Languages, compilers, and tools for embedded systems**, Volume 34 Issue 7

Full text available: pdf(907.45 KB) Additional Information: full citation, abstract, references, index terms

The SpecInt95 benchmark suite is often used to evaluate the performance of programming tools, including those used for embedded systems programming. Embedded applications, however, are often targeting 8- or 16-bit processors with limited functionality, whereas SpecInt95 has no particular target architecture and a bias towards 32-bit systems. Hence, there are reasons to question the use of SpecInt95 for the evaluation of tools for embedded systems.We present a comparative study of the static prop ...

11 How I feed my cats with Linux

Chris McAvoy

January 2005 **Linux Journal**, Volume 2005 Issue 129

Full text available: html(20.16 KB) Additional Information: full citation, abstract, index terms

Why stay home to feed the cats when you have the Internet, a Linux box and some handy hardware?

12 The *nesC* language: A holistic approach to networked embedded systems

David Gay, Philip Levis, Robert von Behren, Matt Welsh, Eric Brewer, David Culler

May 2003 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation**, Volume 38 Issue 5

Full text available: pdf(177.98 KB) Additional Information: full citation, abstract, references, citings, index terms

We present *nesC*, a programming language for networked embedded systems that represent a new design space for application developers. An example of a networked embedded system is a sensor network, which consists of (potentially) thousands of tiny,

low-power "motes," each of which execute concurrent, reactive programs that must operate with severe memory and power constraints.nesC's contribution is to support the special needs of this domain by exposing a programming model that incorporates ...

Keywords: C, TinyOS, components, concurrency, data races, first-order, modules, nesC, programming languages

13 Integrating OO concepts into a CS0 course

Chuck Pheatt

April 2004 **Journal of Computing Sciences in Colleges**, Volume 19 Issue 4

Full text available: pdf(40.56 KB) Additional Information: full citation, abstract, references

This paper describes the use of field programmable integrated circuits (FPIC) in introducing object oriented (OO) programming concepts into a CS0 Course. Using a low cost device known as the OOPIC (Object-Oriented Programmable Integrated Circuits), students can easily control hardware circuitry while being exposed to OO programming concepts. The OOPIC device simplifies hardware programming by allowing students to use common programming languages (Visual Basic, Java, or C) and a simple developmen ...

Keywords: CS0, OOPIC, field programmable integrated circuits

14 A performance analysis of PIM, stream processing, and tiled processing on memory-intensive signal processing kernels

Jinwoo Suh, Eun-Gyu Kim, Stephen P. Crago, Lakshmi Srinivasan, Matthew C. French

May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture**, Volume 31 Issue 2

Full text available: pdf(239.50 KB) Additional Information: full citation, abstract, references

Trends in microprocessors of increasing die size and clock speed and decreasing feature sizes have fueled rapidly increasing performance. However, the limited improvements in DRAM latency and bandwidth and diminishing returns of increasing superscalar ILP and cache sizes have led to the proposal of new microprocessor architectures that implement processor-in- memory, stream processing, and tiled processing. Each architecture is typically evaluated separately and compared to a baseline architectu ...

15 Ray tracing vs. scan conversion: Comparing Reyes and OpenGL on a stream architecture

John D. Owens, Brucek Khailany, Brian Towles, William J. Dally

September 2002 **Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware**

Full text available: pdf(136.72 KB) Additional Information: full citation, abstract, references, citings, index terms

The OpenGL and Reyes rendering pipelines each render complex scenes from similar scene descriptions but differ in their internal pipeline organizations. While the OpenGL organization has dominated hardware architectures over the past twenty years, a Reyes organization differs in several important ways from OpenGL, including a shader coordinate system that supports coherent texture accesses, a single shader in the vertex stage, and tessellation and sampling instead of triangle rasterization.Hardw ...

16 Empirical comparison of software-based error detection and correction techniques for embedded systems

Royan H. L. Ong, Michael J. Pont

April 2001 **Proceedings of the ninth international symposium on Hardware/software codesign**

Full text available: pdf(463.82 KB) Additional Information: full citation, abstract, references, index terms

"Function Tokens" and "NOP Fills" are two methods proposed by various authors to deal

with Instruction Pointer corruption in microcontrollers, especially in the presence of high electromagnetic interference levels. An empirical analysis to assess and compare these two techniques is presented in this paper.

Two main conclusions are drawn: [1] NOP Fills are a powerful technique for improving the reliability of embedded applications in the presence of EMI, and [2] ...

Keywords: EMI, NOP fill, electromagnetic interference, embedded systems, function token, instruction pointer corruption, software-based error detection techniques

17 Compiler analysis and optimization: Providing time- and space- efficient procedure calls for asynchronous software thread integration

Vasanth Asokan, Alexander G. Dean

September 2004 **Proceedings of the 2004 international conference on Compilers, architecture, and synthesis for embedded systems**

Full text available: pdf(289.56 KB) Additional Information: full citation, abstract, references, index terms

Asynchronous Software Thread Integration (ASTI) provides fine-grain concurrency in real-time threads by statically scheduling (integrating) code from primary threads into secondary threads, reducing the context switching needed and allowing recovery of fine-grain idle time. Unlike STI, ASTI allows asynchronous thread progress.Current ASTI techniques do not support procedure calls in the secondary thread because they lead to timing conflicts during static scheduling. ASTI requires knowing the sec ...

Keywords: asynchronous software thread integration, fine-grain concurrency, hardware to software migration, software-implemented communication protocol controllers

18 Microprocessor applications in the nuclear industry

C. Dwayne Ethiridge

April 1980 **ACM SIGCAS Computers and Society**, Volume 10 Issue 3-4

Full text available: pdf(986.50 KB) Additional Information: full citation, abstract, references

Microprocessors in the nuclear industry, particularly at the los Al amos Scientific Laboratory, have been and are being utilized in a wide variety of applications ranging from data acquistion and control for basic physics research to monitoring special nuclear material in long-term storage. Microprocessor systems have been developed to support weapons diagnostics measurements during undergorund weapons testing at the Nevada Test Site. Multiple single-component microcomputers are now controlling ...

19 Mixed-signal design and simulation: A 16-bit mixed-signal microsystem with integrated CMOS-MEMS clock reference

Robert M. Senger, Eric D. Marsman, Michael S. McCorquodale, Fadi H. Gebara, Keith L. Kraver, Matthew R. Guthaus, Richard B. Brown

June 2003 **Proceedings of the 40th conference on Design automation**

Full text available: pdf(793.60 KB) Additional Information: full citation, abstract, references, citings, index terms

In this work, we report on an unprecedented design where digital, analog, and MEMS technologies are combined to realize a general-purpose single-chip CMOS microsystem. The convergence of these technologies has enabled the development of a low power, portable microinstrument ideally suited for controlling environmental and bio-implantable sensors.

Keywords: ADC, MEMS, PGA, SD, SoC, clock generation, design methodology, inductor, low power, low voltage analog, microcontroller, microsystem, mixed-signal, system-on-chip, varactor

20 Formal languages: Towards direct execution of esterel programs on reactive processors

Partha S. Roop, Zoran Salcic, M.W. Sajeewa Dayaratne

September 2004 **Proceedings of the fourth ACM international conference on Embedded software**

Full text available: pdf(269.22 KB) Additional Information: full citation, abstract, references, index terms

Esterel is a system-level language for the modelling, verification and synthesis of control dominated (reactive) embedded systems. Existing Esterel compilers generate intermediate C code that is subsequently mapped to a suitable target processor. The generated code emulates the reactive features of the language due to lack of support for these features on traditional processors. The resultant code is thus inefficient and bulky. Therefore, Esterel is not so effective for resource constrained embe ...

Keywords: ARE-Bench Auckland reactive benchmark, direct ESTEREL execution, reactive processor architectures

Results 1 - 20 of 37 Result page: **1** 2 next

Web Results **1** - **10** of about **40,000** for **Microcontroller programmable assembly counter timer design**. (**0.26** seconds)

The Z80 Microprocessor Architecture Interfacing **Programming** and ...
Programmable Timers and **Counters**. Z80 CTC - **Counter/Timer** ... The 8051 **Microcontroller** Architecture, **Programming** and Applications book by Kenneth Ayala ...
www.penram.com/Microprocessor/Z80-Microprocessor.asp - 26k -
Cached - Similar pages

8051 **Microcontroller**: Architecture, **Programming**
The focus is on **programming** the Intel 8051 **microcontroller**, ... **Timer Counter** Interrupts. **Timing**. **Timer** Modes of Operation. **Counting** ...
microcontrollershop.com/ product_info.php?products_id=711 - 30k -
Cached - Similar pages

Embedded Systems **Design** with 8051 **Microcontrollers**
Timing Parallel ports **Timer/counters** Serial port Interrupt system Power reduction modes ... **Programming** the 8051 **microcontroller** serial port ...
microcontrollershop.com/ product_info.php?products_id=707 - 27k -
Cached - Similar pages

Handbook of **Microcontrollers**
The "Handbook of **Microcontrollers**" was chosen as the "Electronics Book of the ... Clock/Oscillators **Program Counter** Arithmetic Logic Unit Watchdog **Timers** ...
www.myke.com/hom-book.htm - 15k - Cached - Similar pages

HC05 Overview
Fully static **designed** RAM; One time **programmable** and windowed EPROM ... These low pin **count**, cost-effective **microcontrollers** have a variety of ...
www.freescale.com/webapp/sps/ site/overview.jsp?nodeId=0162462LcRkqhk - 87k -
Cached - Similar pages

HC12 Overview
... 16-bit **counter** with a **programmable** prescaler, overflow interrupt, ... The SPI allows the **microcontroller** to communicate with peripheral devices such as ...
www.freescale.com/webapp/sps/ site/overview.jsp?nodeId=0162462LcRC5dj - 87k -
Cached - Similar pages

Press Release
a leading supplier of 8-bit **microcontrollers** and specialty nonvolatile memory ... providing a real-time clock, two 8-bit and one 16-bit **counter/timer** ...
www.microchip.com/stellent/idcplg?IdcService=SS_
GET_PAGE&nodeId=2018&mcparam=en013182 - 23k - Cached - Similar pages

Press Release
This new field-**programmable**, 40-pin, EPROM-based, **microcontroller** offers large ... a real-time clock as well as an 8- and two 16-bit **counter/timer** modules. ...
www.microchip.com/stellent/idcplg?IdcService=SS_
GET_PAGE&nodeId=2018&mcparam=en013132 - 24k - Cached - Similar pages
[More results from www.microchip.com]

Xecon Camera Control System
The vacuum enclosure of the physical Camera Head **Assembly** was **designed** and ... **Counter-timer** #2 is used to generate an RTOS interrupt once every 250 ...
www.discover-net.net/~xecon/xep12k.htm - 20k - Cached - Similar pages

Kasra Digital Instruments (KDI) Experts in LabVIEW ...
... **Design**,Engineer, Control, ECG, EMG, EEG, **Counter**, **Timer**, Microprocessor, Connector, HVAC, ... NET, **Programming**, **Program**, **Assembly**, HEX, BIN, *.HEX, *. ...
www.kdi-co.com/ - 16k - Cached - Similar pages

Goooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

Search Results Results **1 - 10** of about **1,170** for **Microcontroller programmable assembly counter timer design** -

1. STMicroelectronics | PSD5XX/ZPSD5XX FAMILY FIELD-**PROGRAMMABLE MICROCONTROLLER** PERIPHERALS - PSD501B1, PSD502B1, ...
   ST Generic Part Numbers: PSD501B1, PSD502B1, PSD503B1, PSD511B1, PSD512B0, PSD512B1, PSD513B1
   www.st.com/stonline/books/ascii/docs/7831.htm - 242k - Cached - More from this site - Save - Block

2. Digital Designing with **Programmable** Logic Devices
   "Digital **Designing** with **Programmable** Logic Devices.", Englewood Cliffs, NJ: Prentice-Hall, Inc. (432 pp.) Table of Contents. Preface Chapter 1. An Introduction to **Programmable** Logic Chapter 2. Synchronous Binary **Counter Design** Chapter 3. ... The MC68HC11 **Microcontroller** Chapter 10 ... O, Pulse Accumulator and **Timer** Functions 9.6.1 Port ...
   www.coe.uncc.edu/~jcarter/ddpld.html - 27k - Cached - More from this site - Save - Block

3. Atmel offers **design** tool for **programmable** SoCs
   ... FPGA **design** tools. The FPGA **programmable** logic and AVR **microcontroller** tool suites ... the processor's program **counter**, memory, registers, and ... includes C and **assembly** code modules that ...
   www.embedded.com/shared/printableArticle.jhtml?articleID=9900544 - 4k - Cached - More from this site - Save - Block

4. EETimes.com - Atmel offers **design** tool for **programmable** SoCs
   EE Times editorial mission is to provide engineers and technical managers with timely, relevant news, analysis and opinion - of both technology and business - to help them keep their fingers on the pulse of the industry. ... FPGA **design** tools. The FPGA **programmable** logic and AVR **microcontroller** tool suites ... the processor's program **counter**, memory, registers, and ...
   www.eetimes.com/story/OEG20020215S0071 - 53k - Cached - More from this site - Save - Block

5. High-Performance **Microcontroller** with CAN Module Product Brief (PDF)
   ... bit/16-bit **timer/counter** with. 8-bit **programmable** prescaler ... • Advanced Error Management Features. Special **Microcontroller** Features: ... • Fully static **design**. • Wide operating voltage ...
   www.melabs.com/devicedata/30614a.pdf - More from this site - Save - Block

6. Embedded.com - Atmel offers **design** tool for **programmable** SoCs
   ... FPGA **design** tools. The FPGA **programmable** logic and AVR **microcontroller** tool suites ... the processor's program **counter**, memory, registers, and ... includes C and **assembly** code modules that ...
   www.embedded.com/story/OEG20020215S0071 - 79k - Cached - More from this site - Save - Block

7. Narrow Band Filter Implementation On A Low Cost **Microcontroller** Issues and Performance (PDF)
   ... in C. or **Assembly**. With on-chip ... 16-bit **Timer/Counter** with Separate Prescaler ... **Programmable** Watchdog **Timer** with Separate On-chip Oscillator. – Analog Comparator. Special **Microcontroller** ...
   www.superkits.net/whitepapers/DSP-OnAn-AVR.pdf - 1000k - View as html - More from this site - Save - Block

8. Embedded Processor and **Microcontroller** primer and FAQ
   ... realtime, sci.electronics.**design**, alt.comp.hardware ... 7) **MICROCONTROLLER** PROGRAMMING LANGUAGES 7.1) Machine/**Assembly** language 7.2 ... A special **timer/counter** (the idle **timer**) causes the ...
   www.faqs.org/faqs/**microcontroller**-faq/primer - More from this site - Save - Block

9. STMicroelectronics | Tutorial Notes on the uPSD32xx
   ... of Flash **programmable** system devices, each with a 8032 **Microcontroller** Core, and ... 14 Figure 3. **Timer/Counter** Mode 0: 13-bit **Counter** ... 25.USB Input/Output **Design** inside the uPSD32xxA ...

www.st.com/stonline/books/ascii/docs/10150.htm - 167k - Cached - More from this site - Save - Block

10. DR8052EX RISC **Microcontroller** (PDF)
... gram **Counter** (PC) and related ... **Programmable** Watchdog **Timer** ... **Design** Experience. The users should be familiar with the 8052 **microcontroller**. family and skilled in C and **assembly** ...
xilinx.com/products/logicore/.../dcd/dcd_dr8052ex_**microcontroller**.pdf - More from this site - Save - Block

Results Page:
1 2 3 4 5 6 7 8 9 10 ▶ **Next**

Web | Images | Video | Directory | Local | News | Products
Your Search: Microcontroller programmable assembly counter timer design Search